

图文 012、案例实战：每日百万交易的支付系统，JVM栈内存与永久代如何设置？

3324 人次阅读 2019-07-12 07:00:00

详情 评论

案例实战

每日百万交易的支付系统，JVM栈内存与永久代大小又该如何设置？

给大家推荐一套质量极高的Java面试训练营课程：



作者是中华石杉，石杉老哥是我之前所在团队的 Leader，骨灰级的技术神牛！

大家可以点击下方链接，了解更多详情，并进行试听：

[21天互联网Java进阶面试训练营（分布式篇）](#)

重要说明：

最近不少同学留言反馈，说希望建立一个微信群，供大家进行JVM专栏的学习交流。

这个提议非常好，不过管理微信群是一件挺费时的事儿，我平时工作较忙，实在抽不出时间来进行群管理。

正好石杉老哥的面试训练营建了微信交流群，并且还请了不少一线大厂的助教。

因此跟石杉老哥商量了一下，决定厚着脸皮“鸠占鹊巢”。购买了我JVM专栏的小伙伴，可以加入石杉老哥的微信群，在群里讨论交流技术。

如何加群，请参见文末（注：如果之前已经加过的，就不要重复加群了）

目录：

- 前文回顾
- 基于案例，说说不合理设置内存的反面示例
- 大促期间，瞬时访问量增加十倍
- 少数请求需要几十秒处理，导致老年代内存占用变大
- 老年代对象越来越多导致频繁垃圾回收
- 反面案例总结
- 如何合理设置永久代大小？
- 如何合理设置栈内存大小
- 昨日思考题分析
- 今日思考题

1、前文回顾

昨天的文章给大家基于案例分析了一下，如果我们准备上线一个新的系统，如何根据这个系统未来预估的业务量，访问量，去推算这个系统每秒种的并发量，然后推算每秒钟的请求对内存空间的占用，进而推算出整个系统运行期间的JVM内存运转模型。

然后基于这个推算出来的JVM内存运转模型，紧接着去在系统上线前就选择一个合理的机器配置，要多大内存的机器，另外给JVM堆内存空间一个合理的大小。

其实这是一项非常基础的技能，因为对于大厂工程师，每次上线一个新的系统，他可能上线就会面临很大的访问压力

所以必须要学会合理预估内存压力，然后选择对应的机器配置，并且给出合理的内存大小，希望大家好好掌握。

2、基于案例，说说不合理设置内存的反面示例

昨天说的是一个正面的例子，即如何合理的设置内存大小。

今天来说一个反面的不合理设置内存大小导致的问题，也是我们之前线上生产系统真实经历过的一次故障，但是当时不是发生在支付系统中的，是另外一个系统。

不过为了便于大家理解，咱们还是以支付系统作为案例来说明好了，其实思路是一样的。

比如现在我们假设一个前提，就是支付系统因为没有经过合理的内存预估，所以直接选用了1台2核4G的虚拟机来部署了线上系统，而且就只用了一台机器

然后线上JVM给的堆内存大小，仅仅就只有1G，扣除老年代之后，新生代其实就几百MB的内存空间，大家看下图。



好了，接着我们还是用昨天的那个业务压力，就是每天100万交易，高峰期每秒大概100笔支付交易，对应核心的支付订单对象有100个创建出来，每个支付订单对象占据500左右的字节大小，总共就是50kb左右。

然后一笔交易要1秒来处理，所以这100个对象在新生代中存在1秒的期间会被人引用，是无法被回收的。

而且我们之前说过一个全局预估的思路，从核心的支付订单对象扩展开来，拓展到系统里其他的对象中去，起码可以把内存占用扩大了10倍~20倍

比如我们就扩大个20倍好了，那么说明1秒之内，总共会创建出来大概1MB左右的对象，无法被回收。

3、大促期间，瞬时访问量增加十倍

其实按照估算出来的内存压力，你这么小的新生代在系统正常运行的情况下，其实还不算什么大问题

因为每秒新增1MB对象，然后几百秒过后，新生代快满了，自然就会触发Minor GC，回收掉里面99%的垃圾对象。

你要是内存那么小，最多就是发现系统每隔几分钟略微卡顿一下，因为这个时候在进行垃圾回收，会影响系统性能

至于为什么影响系统性能，下周垃圾回收的主题里会分析的。

但是现在我们假设，**如果你的电商系统搞大促活动呢？**

一般搞大促活动，很可能导致你的压力瞬间增大10倍，因为平时不来你网站的人，今天都来了。

此时可能会发现，每秒钟你的支付系统不是100笔订单了，可能是每秒钟上千笔订单。

这个时候你的系统压力本身就会很大了，不光是内存，尤其是线程资源、CPU资源，都会几乎打满。内存就更是岌岌可危了。

4、少数请求需要几十秒处理，导致老年代内存占用变大

咱们就针对内存来分析一下。

现在假设你每秒1000笔交易，那么每秒钟系统对内存的占用增加到10MB以上

我们甚至可以再大胆一点，预估每秒对内存占用达到几十MB，甚至上百MB也可以，因为毕竟大促时流量激增，就一切围绕这来预估。

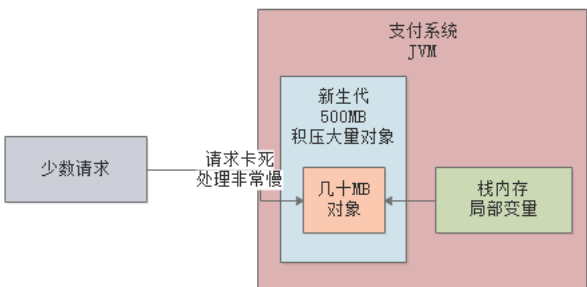
而且最可怕的一点是，可能你每秒过来的1000笔交易，不再是1秒就可以处理完毕了，因为压力骤增，会导致你的系统性能下降，可能偶尔会出现每个请求处理完毕需要几秒钟，甚至几十秒的时间。

此时我们看下图可能出现什么问题，假设你的新生代里已经积压了很多的数据，都快满了。



然后呢，此时内存里有比如几十MB的对象都被人引用着，因为少数请求突然处理的特别慢。

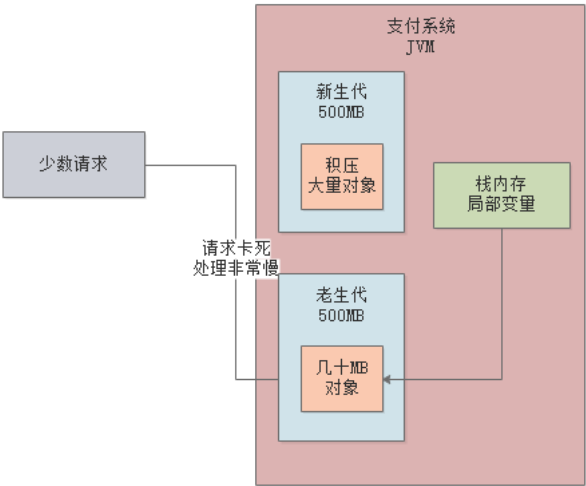
为什么会处理特别慢？因为压力太大，导致系统性能太差了，如下图。



这个时候，如果你要再次在新生代里分配对象，那么是不是会导致一次Minor GC去回收新生代？

没错，但是可能回收掉大量的对象之后，那少数几十MB的对象还在，因为少数请求特别的慢。

然后很快新生代继续被填满，再次触发Minor GC，然后少数几十MB的对象还在，此时多次之后后，就会被转移到老年代去，如下图。



5、老年代对象越来越多导致频繁垃圾回收

那么大家思考一下，上述流程如果反复来多次，就是时不时有少数请求特别慢，创建的对象在新生代反复多次没法被回收，然后就会被弄到老年代去

然后后续处理完之后，老年代里的对象就没人引用了，成为了垃圾对象。

经常重复这个流程，老年代里的垃圾对象，是不是就会越来越多？

一旦老年代的垃圾对象越来越多，迟早会满，然后就会触发老年代的垃圾回收，而且这个老年代被占满的频率还很快，可能会频繁触发老年代的垃圾回收。

大家要知道，老年代的垃圾回收速度是很慢的，这个为什么慢，下周会给大家讲

但是在上述场景下，我们基本可以分析出来，如果你不合理的设置内存，就会导致新生代内存不充足，然后导致很多对象不停的迁移到老年代去，最后导致老年代也要不停的进行垃圾回收。

最后这频繁的垃圾回收，就会极大的影响系统的性能。

6、反面案例总结

本文就通过一个支付系统内存设置过小，然后突发巨大的流量压力，突发的性能抖动，最后导致很多对象长期在新生代被人引用，无法被回收，最后持续进入老年代，最后触发老年代内存都频繁占满，然后老年代都频繁被垃圾回收。

这是我们之前另外一个线上系统真实发生的场景，只不过用支付系统作为案例给大家解释了一遍，所以大家更能从反面体会到，不合理的预估 业务系统压力，不合理的设置内存大小，就可能会导致很大的问题。

7、如何合理设置永久代大小？

话说回来，如何合理设置永久代大小呢？

其实一般永久代刚开始上线一个系统，没太多可以参考的规范，但是一般你设置个几百MB，大体上都是够用的

因为里面主要就是存放一些类的信息，后面也会用专门的案例给大家分析，什么样的系统容易出现永久代内存溢出。

8、如何合理设置栈内存大小

其实这个栈内存大小设置，一般也不会特别的去预估和设置的，一般默认就是比如512KB到1MB，就差不多够了。

这就是每个线程自己的栈内存空间，用来存放线程执行方法期间的各种布局变量的。后面也会用专门的案例演示，栈内存什么时候会发生内存溢出。

9、昨日思考题分析

昨天让大家去思考自己平时负责的系统，有没有按照这个思路去预估业务系统压力，然后给一个合理的内存设置

其实就是希望大家以后建立起来一个全面的工程素养，每个合格的工程师，都应该在上线系统的时候，对系统压力做出预估，然后对JVM内存、磁盘空间大小、网络带宽、数据库压力做出预估，然后各方面都给出合理的配置。

10、今日思考题

看完今天的文章，我想给大家一个思考题，假设大家手头负责的系统业务量暴增100倍，甚至1000倍，按照我们文章的思路，自己去分析一下，有没有可能出现JVM上的问题？

要是业务真的增长这么多，你应该怎么调整机器配置和JVM内存大小？

End

专栏版权归公众号**狸猫技术窝**所有

未经许可不得传播，如有侵权将追究法律责任

如何加群？

- 1、添加微信号：Giotto1245 （微信名：Jarvis）
- 2、发送 Jvm专栏的购买截图
- 3、人工操作，发送截图后请耐心等待被拉群

最后提醒：之前加过面试群的同学就不要重复加了

常见问题解答：

一、如何生成自己的分享海报并获取返现？

方式1：

点击文章右上角**邀请好友**（如下图），生成自己的专属海报。

将海报发送给好友或分享朋友圈，朋友通过扫描你分享的海报购买课程，你将**获取返现24元**，可在个人中心中提现：

累计邀请30人，你将升级为高级推广员，此后每成功邀请一位朋友，返现翻倍。换句话说，从第31人开始，每成功邀请一位朋友，你将**获取返现48元**